

# Table of Contents

|   |          |
|---|----------|
| <b>Acceptance criteria for dependencies.....</b>  | <b>1</b> |
| Necessity.....                                    | 1        |
| Non triviality.....                               | 1        |
| Non duplication.....                              | 1        |
| Usage of standard techniques.....                 | 1        |
| Integration with the gLite software stack.....    | 1        |
| Provision by supported OS.....                    | 1        |
| Multi-platform support.....                       | 1        |
| Availability through a maintained repository..... | 1        |
| Overriding OS dependencies.....                   | 1        |
| Minimal impact.....                               | 2        |
| Stable version.....                               | 2        |
| Track record of ongoing support.....              | 2        |
| License.....                                      | 2        |
| Security.....                                     | 2        |
| Documentation.....                                | 2        |
| Source code.....                                  | 2        |
| Dependencies of dependencies.....                 | 2        |
| After restructuring.....                          | 2        |
| Dependency classification.....                    | 3        |

# Acceptance criteria for dependencies

Aim - reduce the dependency burden of the gLite middleware as far as possible

Each dependency will be evaluated against the following criteria;

## **Necessity**

The dependency is used to enable functionality that is genuinely required in production (has not been obsoleted, is not speculative).

## **Non triviality**

The dependency must supply functionality that cannot be trivially implemented in the dependant application. This includes the possibility of copying selected functions into the gLite codebase.

## **Non duplication**

A new dependency should be only introduced if a similar functionality is not already provided by an already existing dependency used within the gLite software stack or by the gLite software stack itself.

## **Usage of standard techniques**

In the case of 3rd party software, established projects already used within the community will be given priority.

## **Integration with the gLite software stack**

If not supplied by the gLite codebase, the dependency should be able to coexist with any other part of the gLite software stack and its other dependencies, and should not introduce any incompatibilities for other parts of the gLite software stack.

## **Provision by supported OS**

A dependency supplied by all supported Operating Systems is preferred over one supplied by a subset (or none). In this case the dependant application should build and run with the versions natively supplied by the supported platforms.

## **Multi-platform support**

For 3rd party software, it must be available and supported on all gLite targeted platforms. For the moment the main gLite target platforms are SL(C)3 and SL(C)4/i386, SL(C)4/x86\_64 and Debian4.0/i386.

## **Availability through a maintained repository**

3rd party dependencies supplied via a well maintained repository compatible with the target OSs are preferred (eg jpackage).

## **Overriding OS dependencies**

Dependencies may be represented in the supported platforms, but at lower versions, or without critical patches. This necessitates provision of such packages for the affected OSs, so requires strong justification.

## **Minimal impact**

The impact of the introduction of a new 3rd party software should be evaluated in terms of size and functionality. This concerns e.g. cases where only a small functionality of the new software is used while a huge (set of) libraries needs to be included.

## **Stable version**

The 3rd party software used for the gLite software should be a stable, released version and should not be a beta or development version.

## **Track record of ongoing support**

In the case of a 3rd party dependency, the provider has to officially support the version in question.

## **License**

The license of the external software must be compatible with the gLite software. The dependency must be freely distributable by the gLite project. Exceptions to this should be carefully studied.

## **Security**

All 3rd party software the gLite software must meet some minimum security requirements;

- There have been no published security vulnerabilities that have not been fixed in the versions the gLite software stack is using.
- No passwords are viewable in plain text nor are passwords kept in an open-access file or database in plain text. Plain text may be used in databases or files that can otherwise be secured from general users.
- Software is written in a manner that promotes the highest security state applicable for this software.
- Software uses Authentication and Access control as appropriate.
- Support of the software must show the ability to handle any upcoming security issues in a reasonable time scale.

## **Documentation**

Appropriate documentation for the 3rd software must be available.

## **Source code**

For 3rd party dependencies, availability of the source code is preferential, however not strictly required if all necessary platforms have available binaries.

## **Dependencies of dependencies**

If a proposed dependency has itself dependencies on other packages, the ensemble will be assessed according to the criteria listed here.

## **After restructuring**

These criteria will continue to be applied for assessing new dependencies. Introducing any dependency should be carefully evaluated BEFORE starting any development. The final agreement should be given by the EMT by submitting a request describing

- use case
- description
- download location
- commenting the individual acceptance criteria

Enough time (in general two weeks) should be given to the integration team as well as by the other software clusters to evaluate the addition.

## Dependency classification

Worth considering?? The acceptance criteria are stricter as you descend the list

- A supplied by gLite codebase
- B supplied by all supported OSs at the correct version
- C supplied by a 3rd party, or requires an update to packages supplied by some or all supported OSs

-- JoachimFlammer - 25 Jun 2007

---

This topic: EGEE > ExternalSoftwareAcceptanceCriteria

Topic revision: r6 - 2007-06-26 - OliverKeeble



Copyright &© by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Ask a support question or Send feedback